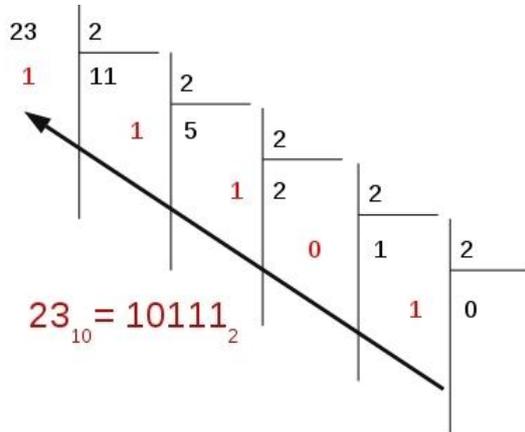


1. Conversion décimale vers binaire

Pour convertir un nombre entier positif N de la base décimale à la base binaire, il faut opérer par des divisions successives du nombre N par 2. Les restes des divisions constituent la représentation binaire.

Ex :



On arrête les divisions par deux quand le quotient devient nul.

On lit alors les reste de division dans l'ordre inverse de leur calcul.

Pour arriver au dernier reste on a effectué n division d'où le terme puissance croissante.

1.1. Conversion décimale vers binaire

Compléter la fonction « `conversion_dec_bin` » .

```
# coding: utf-8
```

```
def conversion_dec_bin(n):
```

```
    """
```

```
    On calcul les différents restes de division par 2
    du nombre tant que le quotient n'est pas nul
    """
```

```
    ch = ""
```

```
    #Cas particulier où le nombre est égal à 0
```

```
    if : ??????????????????????
```

```
        ch = ??????????????????????
```

```
    else :
```

```
        # Boucle répétitive avec condition
```

```
            #calcul du quotient
```

```
            # calcul du reste
```

```
            # condition d'évolution
```

```
            ch = # creation de la chaine dans le bon ordre
```

```
    return ch
```

```
print(conversion_dec_bin(0))
```

```
print(conversion_dec_bin(1))
```

```
print(conversion_dec_bin(11))
```

```
print(conversion_dec_bin(255))
```

```
0
```

```
1
```

```
1011
```

```
11111111
```

2. Conversion décimale vers hexadécimal

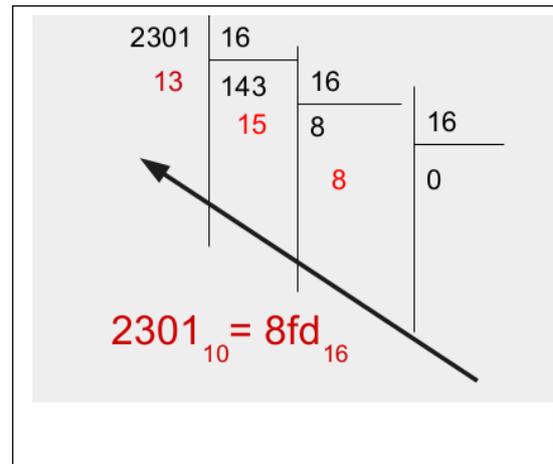
S'inspirer du binaire pour la conversion décimale vers hexadécimal pour écrire une fonction `conversion_dec_hex`. Elle converti un nombre entier vers une chaîne de caractère **représentant sa conversion en hexadécimal**. On utilisera avec profit la fonction suivante de conversion pour un seul nombre hexadécimal :

```
def dec_vers_hexa(n):
```

```

    if n == 10:
        return 'A'
    elif n == 11:
        return 'B'
    elif n == 12:
        return 'C'
    elif n == 13:
        return 'D'
    elif n == 14:
        return 'E'
    elif n == 15:
        return 'F'
    else:
        return str(n)

```



Vous devez trouver

```

print(conversion_dec_hex(0))    0
print(conversion_dec_hex(15))   F
print(conversion_dec_hex(16))   10
print(conversion_dec_hex(255))  FF

```

[Tapez ici]

[Tapez ici] [Tapez ici]

3. Caractères ASCII

En vous inspirant du cours : `ord()` renvoie la valeur unicode d'un caractère
l'unicode commence par les caractères ASCII qui gardent leurs valeurs
`chr()` renvoie le caractère dont on donne son code

```
"""
```

```
c = 'p'
```

```
print(f"La valeur ASCII du caractère {c} est", ord(c))
```

```
print(f"Le caractère ASCII dont le code est {ord(c)} est le caractère ",  
chr(ord(c)))
```

Faites afficher un tableau avec les caractères ASCII du numéro 48 à 127 grace à la fonction ASCII

```
def tableau_ASCII(depart, fin):  
    print("Caractère | code | binaire | hexadécimal ")
```

A compléter

```
tableau_ASCII(48,127)
```

Affiche :

Caractère	code	binaire	hexadécimal
0	48	110000	30
1	49	110001	31
2	50	110010	32

Dans un premier temps contentez vous de faire afficher les valeurs ligne à ligne. Dans un deuxième temps préoccupez-vous de l'alignement avec la documentation suivante :

Les colonnes mesurent 10 caractères pour aligner les caractères on utilise la documentation suivante

Alignement <https://courspython.com/print-format.html>

Par exemple, il est possible d'indiquer un nombre entier qui précise la largeur à utiliser pour afficher la valeur transmise. Il y aura ainsi au sein de la chaîne produite un nombre de caractères fixé pour afficher la valeur. Si la valeur à une longueur trop petite, des espaces seront ajoutés pour arriver à la largeur voulue.

[Tapez ici]

[Tapez ici] [Tapez ici]

Exemple

```
print(f"Les valeurs sont {a:10} et {b:10}")
Les valeurs sont          5 et          8
```

Ici les valeurs de *a* et *b* sont affichées en occupant 10 caractères chacune.

Il est possible d'aligner à gauche en utilisant le signe inférieur <.

```
print(f"Les valeurs sont {a:<10} et {b:<10}")
Les valeurs sont 5          et 8
```

Pour centrer, on utilise le caractère ^.

```
print(f"Les valeurs sont {a:^10} et {b:^10}")
Les valeurs sont      5          et      8
```

Quelle relation existe-t-il entre les minuscules et les majuscules ?

4. Caractères Unicode

Ouvrez le site <https://unicode-table.com/fr/sets/greek-symbols/>

Faites afficher ligne par ligne les différents caractères grecs

Méthode :

- prendre le code hexadécimal de
- prendre le code de la dernière lettre de l'alphabet

Convertir en décimal les deux valeurs pour obtenir l'intervalle des lettres. de façon à utiliser une boucle en pour. Regarder fonction chr()

Faites afficher :

Caractère	Unicode
A	U0391
B	U0392
Γ	U0393
...	...

Α

A Lettre majuscule grecque Alpha

U+0391

ω

ω Lettre minuscule grecque Oméga

U+03C9